**PLANNING A COMPLEX ALGORITHM**

**DESIGN THE ROUTINE**

CHECK PREREQUISITES

Define the problem

*Match Results Produced from our S1 program does not contain any formattable properties. There is also no page structure*

Information the routine will hide

*All previously calculated routines will remain the same and work in the background*

Inputs to the routine

*-*

Outputs from the routine

*Displays Text in Separate Divisions*

Pre-conditions

*All Data is calculated and inputted. Some Document Structure has to be set up with*

*appropriate Divisions and sections.*

Post-conditions

*Data is now displayed on the screen from dynamic generation with html dom elements*

Name the Routine

*Returning Match Results from The Commonwealth Games 2018*

Decide how to test the routine

*We will be looking at the console and at the page structure generated in the Inspect Element. If the routine is successful, we should see structure generated Dynamically not already set out in the page document, E.g new paragraphs with the info. This should also be able to be styled using CSS in our document library (To be added)*

Research functionality available in standard libraries

*The majority of this assignment is working with HTML DOM, which is readily available within JS Libraries. I will have to Learn how to create new DIVs, Headings, Paragraphs and also be able to assign new attributes to them for styling Later on*

Think about error handling

*Generally, with this dynamic generation Physical errors are able to be identified and fixed as they will not be visible on the page.*

Think about efficiency

*Looping will be used to generate the majority of the data. This approach will avoid the Hard Coding of every single result.*

Research algorithms & data types

*Dom Elements have a different syntax of assigning variables and attributes than regular html. Assigning them to a variable then using functions to assign these attributes seems to be the simplest way of doing it.*

**WRITE PSEUDOCODE**

1. Think about the data
2. Check the pseudocode
3. Try ideas in pseudocode

**CODE THE ROUTINE**

1. Write the declaration
2. Turn pseudocode into comments
3. Fill in code below comments
4. Check if code can be factored

**CHECK THE CODE**

1. Mentally check for errors
2. Step through in Debugger
3. Test the code
4. Remove errors in the code
5. Clean up